

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Absolvování individuální odborné praxe

Individual Professional Practice in the Company

Zadání bakalářské práce

Student: **Pavel Vičar**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: Absolvování individuální odborné praxe
Individual Professional Practice in the Company

Jazyk vypracování: čeština

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: E LINKX a.s.
2. Struktura závěrečné zprávy:
 - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
 - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
 - c) Zvolený postup řešení zadaných úkolů.
 - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
 - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
 - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.


Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **doc. RNDr. Petr Šaloun, Ph.D.**


Konzultant bakalářské práce: Ing. Roman Hrdý

Datum zadání: 01.09.2016

Datum odevzdání: 14.07.2017


doc. Ing. Jan Platoš, Ph.D.
vedoucí katedry




prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Souhlasím se zveřejněním této bakalářské/diplomové práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských/magisterských programech VŠB-TU Ostrava.

V Ostravě 12. července 2017

.....

Abstrakt

Tato bakalářská práce popisuje průběh praxe ve firmě E LINX a. s. V první části krátce popisují firmu a důvod, proč jsem si zvolil odbornou praxi jako téma bakalářské práce. V hlavní části popisují projekty, na kterých jsem se podílel, průběh řešení zadané problematiky a technologie, které jsem při tom využil. V další části popisují uplatněné znalosti ze studia a také znalosti, které jsem postrádal a celkové zhodnocení průběhu praxe.

Klíčová slova: Odborná praxe, C#, Elinkx

Abstract

This bachelor thesis describes process of my professional practice in company E LINKX a. s. In the first part I introduce the company and reason why i chose professional practice as topic of my bachelor thesis. In the mai part I describe projects I've worked on, process of solution of given issues and technologies which I have used. In next section I describe application of knowledge acquired during studies and also knowledge I have lacked and overall evaluation of professioal practice.

Key Words: Professional practice, C#, Elinkx

Obsah

Seznam použitých zkratk a symbolů	7
Seznam obrázků	8
1 Úvod	9
2 O firmě E LINX	10
3 Pracovní zařazení	11
4 Školení	12
4.1 Objektově orientované programování	12
4.2 Návrhové vzory	12
4.3 Sql	12
5 Projekty	13
5.1 Strom	13
5.2 Wizard	14
5.3 Služba certifikátů	15
5.4 Zpracování dat	18
6 Časové náročnosti	23
7 Využité a chybějící znalosti	24
7.1 Využité znalosti	24
7.2 Chybějící znalosti	24
8 Závěr	25

Seznam použitých zkratek a symbolů

WF	– Windows forms aplikace
Dll	– Dynamicky připojovaná knihovna
API	– Rozhraní pro programování aplikací
CSV	– Souborový formát, jehož položky jsou odděleny čárkami nebo středníky
MVC	– Model-view-controller
Linq	– Language integrated query
AJAX	– Asynchronní zpracování webových stránek
REST	– Representational state transfer
JSON	– Javascript Object Notation

Seznam obrázků

1	Vstupní data wizardu	14
2	Wizard	15
3	Detail certifikátu	17
4	Zpracování dat počáteční verze webového rozhraní	22

1 Úvod

Při volbě bakalářské práce jsem se rozhodl absolvovat odbornou praxi, motivem mé volby bylo zjistit, zda jsou mé znalosti nabyté během studia použitelné k výkonu práce ve firmě a při řešení reálných problémů. Hlavním motivem však bylo získat zkušenosti práce ve firmě ve svém oboru. Odbornou praxi jsem absolvoval ve firmě E LINX a. s. O firmě jsem se dozvěděl skrze systém katis, kde byly zobrazeny pozice nabídky praxí. Podrobnější informace jsem si našel na internetových stránkách firmy.

V této bakalářské práci nejdříve popisuji firmu, ve které odbornou praxi absolvoval. V hlavní části popisuji jednotlivé projekty, na kterých jsem se podílel. V další části popisuji použité technologie během praxe. V závěru zhodnotím uplatnění znalostí získaných při studiu a znalosti získané při vykonávání odborné praxe.

2 O firmě E LINX

Česká společnost působící na trhu více než patnáct let. Elinx je firma zabývající se vývojem ekonomického informačního systému (ERP), vývoje softwaru pro e-business a řešení pro distribuci a logiku. Informační systémy zpracovává na platformě Microsoft (.NET, Microsoft Sql Server, Win Forms a Windows Service). Společnost je členem Microsoft Partner Network a dlouholetým držitelem Gold Hosting. Tento certifikát udílí společnost svým nejvýznamnějším obchodním partnerům, čímž potvrzuje vysokou míru odbornosti a důležité postavení na trhu. Společnost se také řadí mezi významné IT partnery s evropským přesahem.

3 Pracovní zařazení

Po výběru firmy v systému katis mě firma kontaktovala a pozvala na přijímací pohovor. Na pohovoru byly prověřeny moje odborné znalosti programovacího jazyka C#. Dále byly prověřeny schopnosti mého uvažování, schopnosti podívat se na problém z více úhlů a schopnost analýzy a řešení problému. Na konci mi byl zadán jednoduchý úkol k vypracování jako součást přijímacího procesu.

Po přijímacím pohovoru jsem byl zařazen do týmu studentů, kteří se přihlásili do firmy na odbornou praxi již dříve. S nimi jsem pracoval na projektu pod názvem Wizard. Po dokončení wizardu jsme dostali každý individuální úkol, po jeho vypracování jsem byl přiřazen ke kolegovi, který se také přihlásil na praxi, se kterým jsem pracoval po zbytek praxe.

4 Školení

Po přijetí na praxi jsem se účastnil několika školení. Školení se pohybovaly v okruhu programování a databází. Školení probíhala formou prezentací z naší strany, strany studentů, dostali jsme zadání nebo téma, jehož problematiku jsme si měli nastudovat, následně jsme si ze získaných znalostí měli připravit prezentaci. Po nastudování a prezentování proběhla diskuze, při které nám byly pokládány otázky, a celkově jsme probrali danou problematiku. Součástí prezentací byly také praktické ukázky.

4.1 Objektově orientované programování

První školení, které jsem absolvoval. Na školení se probíralo téma objektově orientovaného programování, základní principy objektů, vztahů mezi nimi, vlastnosti jako jsou dědičnost, zapouzdření a další. Víceméně se opakovali teoretické znalosti, které jsem již slyšel v předmětech Programování I a II.

4.2 Návrhové vzory

Na školení jsme probírali návrhové vzory, bylo nám vysvětleno obecně, k čemu se návrhové vzory používají, kdy by se měli návrhové vzory použít a vysvětlili jsme si konkrétní návrhové vzory. Součástí školení byly také praktické ukázky, ukázky zdrojového kódu implementovaného vzoru a jeho vysvětlení. S tímto tématem jsem se při studiu setkal v předmětu Vývoj informačních systémů.

4.3 Sql

Na tomto školení se probírali téma databází. Probrali jsme relační databáze obecně. Tabulky a způsob uložení dat, filtrace řádků, výběry sloupců v selectu s ohledem na výkon a vytížení databáze. Primární a cizí klíče, proč a k čemu slouží a jak zajišťují integritu dat. Další na seznamu byly datové typy, základní typy jako jsou INT, BOOL a další, binární datové typy, rozdíly mezi datovými typy, vliv na výkon, konverze mezi nimi. Téma spojování tabulek bylo další na seznamu, byly probrány varianty příkazu join, množinové operace nad řádky a praktické ukázky spojování tabulek včetně špatně napsaného joinu, praktické filtrování řádků na NULL a NOT NULL hodnoty. Víceméně to byla opět teorie, se kterou jsem se již při studiu setkal v předmětech věnujících se databázím.

5 Projekty

V této části se budu zabývat projekty, na kterých jsem se podílel. Přiblížil bych jejich základní funkčnost, jaké byly požadavky na projekt, na jaké problémy jsem narazil a jaký jsem zvolil postup při řešení problému.

5.1 Strom

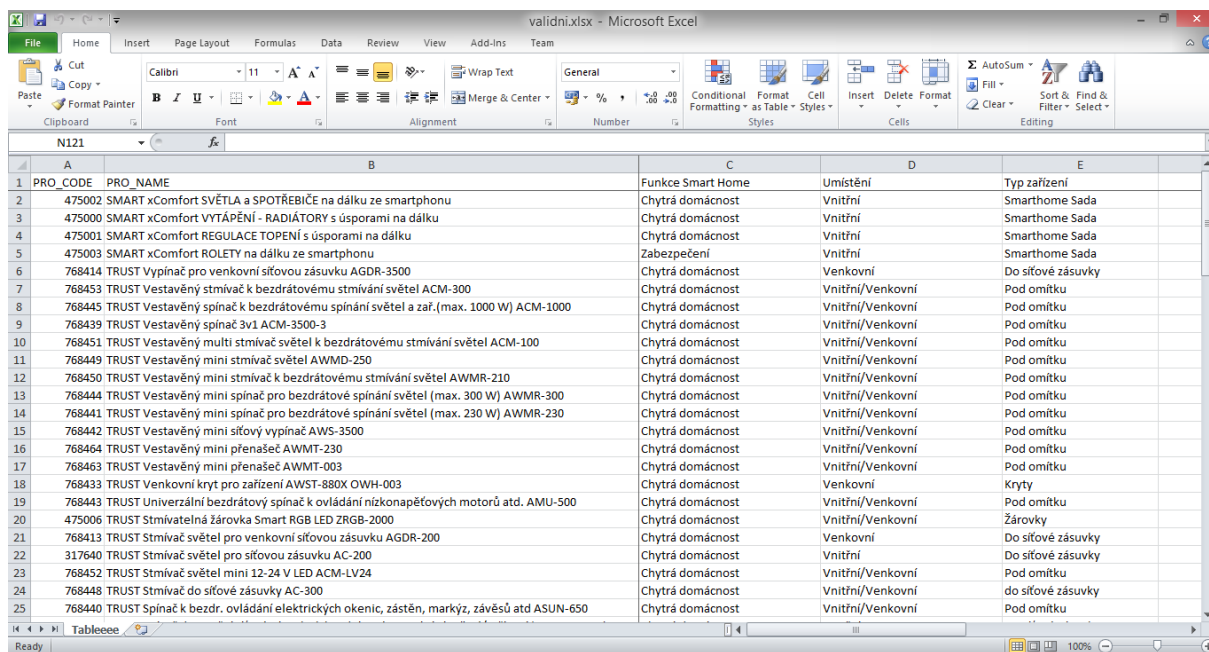
Projekt, který mi byl zadán jako součást přijímacího protokolu. Projekt musel být naprogramován v jazyce C#, v kódu mohli být použity pouze metody dostupné v .NET frameworku, metody z jiných knihoven bylo zakázáno použít. Výsledkem zadání měl být tedy projekt obsahující třídu, která měla reprezentovat strom. Uzly stromu musí umožňovat přidávání a odebírání podřízených uzlů, každý uzel obsahuje i data, třída tedy byla generická. Další úkol byl vytvořit algoritmus, který strom sekvenčně projde a vypíše data všech listů na samostatných řádcích a popsat problémy k jakým by mohlo při provádění algoritmu dojít.

Třidu jsem implementoval, zadání bylo celkem jednoduché, pro výpis jednotlivých jsem implementoval rekurzivní metodu, která vypíše hodnotu uzlu, u kterého byla metoda zavolána, následně zavolá tuto metodu u svých potomků a předá jim parametr offset + 1, který udává odsazení řádku od kraje. K problémům jaké by mohli nastat, jediný problém, který mě napadl, byl problém souběhu, kdy při výpisu dat by se ke stromu pokusilo přistoupit další vlákno, tento problém by se dal jednoduše vyřešit příkazem lock.

5.2 Wizard

Wizard je první projekt, na kterém jsem se podílel. K projektu jsem byl přiřazen ihned po přijetí na praxi. Cílem projektu bylo vytvořit webovou aplikaci za použití technologie .NET a Microsoft SQL Serveru. Aplikace sloužila pro správu wizardů. Wizard obsahuje seznam produktů a posloupnost kroků. Kroky jsou na sobě navzájem závislé, uživatel volí parametry a podle zvolených parametrů se na konci filtrují produkty odpovídající zvoleným kritériím.

Data wizardu jsou obsažena v excel souboru, při vkládání je nutné validovat, zda excel obsahuje několik počátečních sloupců povinných dat, to jsou data o produktu. Následující sloupce jsou kroky, které wizard obsahuje.



PRO_CODE	PRO_NAME	Funkce Smart Home	Umístění	Typ zařízení
475002	SMART xComfort SVĚTLA a SPOTŘEBIČE na dálku ze smartphonu	Chytrá domácnost	Vnitřní	Smarthome Sada
475000	SMART xComfort VYTÁPĚNÍ - RADIÁTORY s úsporami na dálku	Chytrá domácnost	Vnitřní	Smarthome Sada
475001	SMART xComfort REGULACE TOPENÍ s úsporami na dálku	Chytrá domácnost	Vnitřní	Smarthome Sada
475003	SMART xComfort ROLETY na dálku ze smartphonu	Zabezpečení	Vnitřní	Smarthome Sada
768414	TRUST Vypínač pro venkovní síťovou zásuvku AGDR-3500	Chytrá domácnost	Venkovní	Do síťové zásuvky
768453	TRUST Vestavěný stmívač k bezdrátovému stmívání světel ACM-300	Chytrá domácnost	Vnitřní/Venkovní	Pod omítku
768445	TRUST Vestavěný spínač k bezdrátovému spínání světel a zař.(max. 1000 W) ACM-1000	Chytrá domácnost	Vnitřní/Venkovní	Pod omítku
768439	TRUST Vestavěný spínač 3v1 ACM-3500-3	Chytrá domácnost	Vnitřní/Venkovní	Pod omítku
768451	TRUST Vestavěný multi stmívač světel k bezdrátovému stmívání světel ACM-100	Chytrá domácnost	Vnitřní/Venkovní	Pod omítku
768449	TRUST Vestavěný mini stmívač světel AWMD-250	Chytrá domácnost	Vnitřní/Venkovní	Pod omítku
768450	TRUST Vestavěný mini stmívač k bezdrátovému stmívání světel AWMR-210	Chytrá domácnost	Vnitřní/Venkovní	Pod omítku
768444	TRUST Vestavěný mini spínač pro bezdrátové spínání světel (max. 300 W) AWMR-300	Chytrá domácnost	Vnitřní/Venkovní	Pod omítku
768441	TRUST Vestavěný mini spínač pro bezdrátové spínání světel (max. 230 W) AWMR-230	Chytrá domácnost	Vnitřní/Venkovní	Pod omítku
768442	TRUST Vestavěný mini síťový vypínač AWS-3500	Chytrá domácnost	Vnitřní/Venkovní	Pod omítku
768464	TRUST Vestavěný mini přenašeč AWMT-230	Chytrá domácnost	Vnitřní/Venkovní	Pod omítku
768463	TRUST Vestavěný mini přenašeč AWMT-003	Chytrá domácnost	Vnitřní/Venkovní	Pod omítku
768433	TRUST Venkovní kryt pro zařízení AWST-880X OWH-003	Chytrá domácnost	Venkovní	Kryty
768443	TRUST Univerzální bezdrátový spínač k ovládání nízkonapěťových motorů atd. AMU-500	Chytrá domácnost	Vnitřní/Venkovní	Pod omítku
475006	TRUST Stmívatelná žárovka Smart RGB LED ZRGB-2000	Chytrá domácnost	Vnitřní/Venkovní	Žárovky
768413	TRUST Stmívač světel pro venkovní síťovou zásuvku AGDR-200	Chytrá domácnost	Venkovní	Do síťové zásuvky
317640	TRUST Stmívač světel pro síťovou zásuvku AC-200	Chytrá domácnost	Vnitřní	Do síťové zásuvky
768452	TRUST Stmívač světel mini 12-24 V LED ACM-LV24	Chytrá domácnost	Vnitřní/Venkovní	Pod omítku
768448	TRUST Stmívač do síťové zásuvky AC-300	Chytrá domácnost	Vnitřní/Venkovní	do síťové zásuvky
768440	TRUST Spínač k bezdr. ovládání elektrických okenic, zástěn, markýz, závěsů atd ASUN-650	Chytrá domácnost	Vnitřní/Venkovní	Pod omítku

Obrázek 1: Vstupní data wizardu

Správce wizardů

Administrační rozhraní pro správu wizardů.

Wizardy můžete upravit či smazat, případně přidat excel soubor s daty. Můžete také spustit demo wizardu.

Vyhledat wizardy podle parametrů

Obsahuje text:

Obsahuje excel data: ☒ Nezáleží ☐ Ano ☐ Ne

Vyhledat

Seznam wizardů

Název	Popis	Excel	
Validni	hhhh	<input checked="" type="checkbox"/>	🔗 ✕ 📄 ▶ +
ValidONE		<input checked="" type="checkbox"/>	🔗 ✕ 📄 ▶ +

Nový wizard

Obrázek 2: Wizard

5.3 Služba certifikátů

Požadavky na aplikaci byly následovné, vytvořit jednoduchou aplikaci, která bude umožňovat ukládat certifikáty do certifikačního úložiště v systémech Windows 7, 8, 10, kontrolovat dobu jejich vypršení platnosti a poskytovat základní údaje jako jsou veřejný klíč, otisk, dobu platnosti, veškeré operace musí být logovány pro případnou zpětnou kontrolu.

Nebylo specifikováno, jestli aplikace má být WF, webová aplikace nebo konzolová aplikace. Rozhodl jsem se použít WF, konzolovou aplikaci jsem vynechal, protože musíme brát v potaz, že aplikaci bude používat klasický uživatel, který by si s konzolí nedokázal poradit, webová aplikace má své výhody například, můžete se k ní dostat z jakéhokoli zařízení, které má webový prohlížeč, ale z důvodů časového omezení a ne zkušenosti s vývojem webových aplikací jsem zvolil WF, se kterým jsem se již mnohokrát setkal.

5.3.1 Přístup k certifikátům

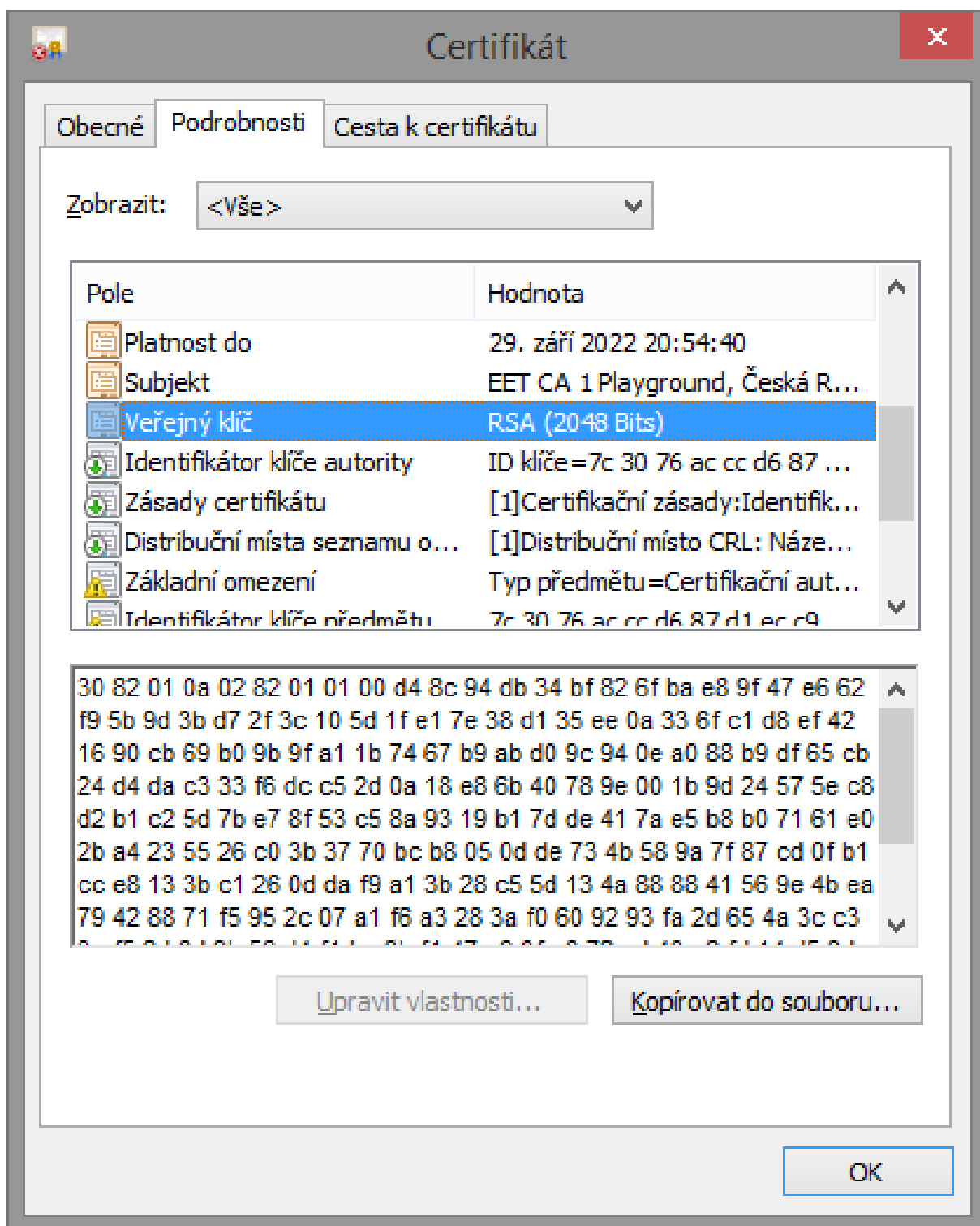
Pro přístup do certifikačního úložiště windows jsem použil třídu dostupnou v .NET frameworku x509certificate2. Tato třída umožňuje přístup do jednotlivých certifikačních úložišť a validaci jejich údajů. Vytvořil jsem si třídu s metodami k přístupu k úložišti, které umožňuje filtrování podle otisku certifikátu, doby platnosti. V této třídě jsou také implementovány metody pro přidávání certifikátů do úložiště, ukládaný certifikát musí být platný, tyto operace jsou dostupné pouze přihlášeným uživatelům a jsou logovány.

5.3.2 Uložiště dat

Pro uložení dat jsem zvolil microsoft sql server, kvůli jeho jednoduchému přístupu v jazyce C#. Pro přístup k databázi jsem využil návrhový vzor table data gateway, kdy jedna třída představuje jednu tabulku a obsahuje veškeré potřebné operace nad tabulkou. Do databáze se ukládají veškerá data, logují se veškeré operace ohledně certifikátů, události se logují dávkově vždy po deseti kvůli snížení nároků na databázi.

5.3.3 Ověřování uživatelů

Ověřování a autorizace uživatelů je pomocí uživatelského jména a heslo, dále se podle rolí určuje, ke kterým funkcím má uživatel přístup. V databázi jsou přihlašovací údaje uloženy, uživatelské jméno jako text a heslo je uloženo jako hashovaná hodnota pomocí algoritmu sha512 a pole salt bajtů. Při vytváření uživatele je k heslu připojeno pole náhodně vygenerovaných bajtů a tato hodnota je poté hashovaná. Při přihlášení se z databáze se vytáhne pole salt bajtů, ty se připojí k zadanému heslu a ověřuje se, zda výsledná hash hodnota je totožná s hash hodnotou v databázi.



Obrázek 3: Detail certifikátu

5.4 Zpracování dat

Jak název napovídá, projekt shromažďuje data z heterogenních zdrojů. Tyto data zpracuje a výsledkem je model, který bude uživateli prezentován v jemu požadované formě. Uživatel požaduje přístup k těmto datům odkudkoliv, požaduje také, aby se mu zobrazil přehled, přehled musí umožňovat filtraci dat podle vybraných sloupců. Uživatel také požaduje export dat, které právě vidí na obrazovce do formátu dokumentu excel csv. Proces shromáždění dat musí být automatický, proces se tedy automaticky spustí ráno každý den. Projekt má dvě hlavní části, první část je program, který se automaticky stará o stažení a import dat, druhá část je tedy webové rozhraní, které umožňuje uživateli přístup k datům a přidávání uživatelských účtů.

5.4.1 Návrh

Podle požadavků měl být vytvořen návrh aplikace, v požadavcích nebylo specifikováno, zda má být aplikace webová nebo klasická okenní aplikace, po domluvě jsme se shodli na tom, že aplikace bude webová z důvodů přístupnosti odkudkoliv, kde je k dispozici internetové připojení. Jelikož aplikace je webová, je nutné zabezpečení, domluvili jsme se, že jednoduché zabezpečení pomocí uživatelského jména a hesla bude pro začátek dostačující.

Na základě informací a požadavků byl vytvořen třídni diagram, konceptuální model databáze a dokument obsahující celkovou analýzu projektu. To jsme po té konzultovali s našim vedoucím, který nám analýzu zhodnotil a navedl nás jakým směrem se dát při implementaci zadání.

5.4.2 Základ aplikace

Při vývoji hlavní aplikace jsme se rozhodli, že pro každý zdroj bude jedna dll knihovna, která bude implementovat přístup a převod dat svého konkrétního zdroje. Tyto knihovny jsou umístěny v definované složce aplikace, která je dynamicky za běhu načítá a spouští. Byl problém s ověřením, protože bylo potřeba, aby třída v knihovně obsahovala konkrétní metody, byly dvě možnosti, první možnost je ručně ověřovat zda třída obsahuje určené metody a poté vytvořit instanci jejího typu, další možnost byla vytvořit knihovnu, která obsahuje rozhraní obsahující předpis všech potřebných metod. Zvolili jsme možnost s rozhraním. V hlavní aplikaci tedy při načítání ověřujeme, zda třída implementuje společné rozhraní a poté vytváříme její instanci. Knihovna s rozhraním také obsahuje třídy reprezentující požadovaný formát dat.

Později byl přidán požadavek na to, aby aplikace uměla podle potřeby spouštět jednotlivé knihovny paralelně. Vytvořil jsem interface, který obsahuje předpis metody pro spuštění, dále dvě třídy reprezentující paralelní a sériový běh, které toto rozhraní implementují. V hlavní třídě poté při vytváření instance se podle hodnoty zapsané v konfiguračním souboru vytváří instance buď jedné, nebo druhé třídy.

5.4.3 Modul pro csv

První modul obsahuje implementaci pro čtení csv souborů. Csv soubory se ukládají na disk, ke kterému má stroj hostující aplikaci přístup. Modul má v konfiguračním souboru specifikováno, kde má csv soubory hledat, při otevření souboru kontroluje, zda má soubor sloupce s požadovaným názvem, pokud soubor obsahuje sloupce navíc, jsou sloupce ignorovány. Výstupem jsou data v požadovaném formátu.

5.4.4 Modul pro excel

Logika průběhu zpracování dat je téměř totožná s modulem pro csv soubory, hlavní rozdíl je v přístupu k datům v souboru, zatím co csv soubor lze jednoduše číst jako text, u excel souborů to není možné. Jako první jsme zkoušeli knihovnu Microsoft Office Interop Excel, která soubor excel dokáže otevřít a pracovat s ním jako s tabulkou a dokáže k jednoduše přistupovat k jednotlivým buňkám a řádkům skrze indexer. Nevýhodou knihovny je, že ke správné funkci potřebuje, aby na hostitelském zařízení byl nainstalovaný Microsoft Office, knihovna využívá totiž samotný excel. Další problém s knihovnou byl, že po skončení algoritmu zůstávaly spuštěné instance programu excel. To se vedoucím příliš nezdálo, a proto bylo nutné modul přepracovat. Jako další způsob modul používá třídu OleDbConnection, která umožňuje přístup k datům v excel souboru obdobně jako k databázi.

5.4.5 Moduly pro xml

Dva moduly obstarávající přístup a zpracování xml souborů, moduly jsou si velice podobné, hlavní rozdíly jsou uložení, ke kterému modul přistupuje. První modul přistupuje k webu za pomoci třídy WebClient, kdy xml soubor je dostupný na www adrese, WebClient jej stáhne jako prostý text. Druhý modul přistupuje na ftp server pomocí přihlašovacích údajů a stahuje soubor do paměti z www response. Data v xml dokumentu jsou ověřeny oproti vytvořenému schématu.

5.4.6 Modul pro webovou službu

Modul přistupuje k externí webové službě, která poskytuje data. Pomocí klienta jsou na server odeslány požadavky s parametry, podle kterých jsou vráceny určitá data. Data jsou ve formátu json, tyto data je nutné ověřit. Ověřování probíhá při převádění na objekty, ověřuje se, zda každý json objekt obsahuje potřebné vlastnosti a zda mají platné hodnoty, například cena nesmí být záporná nebo název produktu nesmí být null. Po převedení vznikne pole objektů, které jsou poté nahrány do databáze.

5.4.7 Webové rozhraní

Portál pro uživatele, zde uživatel vidí importované data, data jsou zobrazeny v tabulce s požadovanými sloupci. Tabulka je stránkovaná, kvůli přehlednosti se zobrazuje vždy jen deset záznamů na stránku. Pro vytvoření webových stránek byly dvě možnosti řešení. Možnosti byly použít starý známý ASP.NET WebForms a nebo novější ASP.NET MVC.

Pro tvorbu bylo zvoleno MVC, protože je novější a modernější než zastaralé ASP.NET. MVC sice postrádá některé ASP prvky, jako například server side komponenty nebo drag and drop návrh stránky, kde může uživatel přetahovat komponenty z lišty přímo na stránku. MVC není tak náročný na vývoj, kód je přehledný, protože odděluje uživatelské rozhraní od logické vrstvy. Webovou část projektu jsem vyvířeli v architektuře ASP.NET MVC. Pro design jsme použili framework bootstrap z důvodu je volně šiřitelný a propracovanosti, bootstrap jsme použili, protože jeho součástí je již přizpůsobivost elementů html pro telefony, což nám ušetřilo hodně času.

Číslování bylo později předěláno na ajax calls, pomocí kterých se obnovuje pouze tabulka s daty, nebylo potřeba, aby se při stisknutí obnovovala celá stránka. Ajax odkazuje na metodu controlleru, která odpoví pouze, pokud je žádost ajax. Metoda metoda vrací pouze html část obsahující tabulku s daty a číslování.

5.4.8 Ověřování uživatelů a zabezpečení

Ověřování uživatelů je pomocí uživatelského jména a hesla, v databázi je heslo uloženo jako hash hodnota, při přihlášení je zadané heslo zahashováno a ověřeno zda je výsledná hodnota totožná s hodnotou uloženou v databázi. Uživatel může vytvářet účty dalším uživatelům. K heslo je převedeno na pole bajtů, ke kterému se poté připojí pole bajtů obsahující náhodně vygenerované hodnoty. T výsledné kolekce se vygeneruje hash. Hash hodnota a pole vygenerovaných bajtů jsou poté uloženy do databáze.

Přihlášený uživatel je uložený v cookies prohlížeče, na straně serveru probíhá ověření v metodě OnAuthenticate, která je specifikována v Global.asax. Tato metoda se volá při každém požadavku na server. K dalšímu ověřování používáme ASP.NET identity a třídu GenericPrincipal, která nám umožňuje přenášet data o přihlášeném uživateli mezi jednotlivými controllery a role přiřazené uživateli, které určují, ke kterým funkcím a metodám webu má uživatel přístup.

Aplikace poběží na IIS, pro zabezpečení používá standartní https protokol a certifikát. V aplikaci jsem využil atributu RequireHttps, díky kterému je pro veškeré požadavky povinný https protokol. Požadavky protokolu http automaticky přesměruje na https.

5.4.9 RESTful web service

Další požadavek, který přišel na projekt po té, co byl již hotový web. Úkolem bylo vytvořit REST webovou službu, která byla napojena na již hotovou databázi s daty. Projekt webové služby má velice podobnou strukturu architektury MVC. Byly vytvořeny metody odpovídajícím požadavkům, které vraceli požadované data ve formátu JSON. Počet vrácených záznamů byl omezen na deset řádků na požadavek a v parametru požadavku muselo být specifikováno číslo stránky.

Naše webové rozhraní bylo přímo napojeno na vrstvu přístupu k databázi, bylo požadováno, aby místo toho web přistupoval k vytvořenému webovému rozhraní a neměl tak přímý přístup k databázi. Bylo proto nutné předělat vnitřní část webu, která přistupovala k databázi tak aby přistupovala a čerpala data s webové služby. Problém byl ten, že web používá pro číslování tabulky modul MvcPagedList, který je rozšířením a převádí IEnumerable na PagedList, ale aby vytvořil správné číslování je potřeba mu předat celou kolekci dat. Proto, aby číselník správně fungoval, bylo potřeba vytvořit třídu, která bude implementovat rozhraní IPagedList a veškerou logiku vytvořit v ní.

Název

Datum od

dd. mm. rrrr

Datum do

dd. mm. rrrr

Filtrovat

Export .csv

pID	pID	Název	Množství	Cena	Měna	Datum
51011	188473	GARMIN Drive 40 Lifetime Czech	10	2590	CZK	22. 12. 2016 0:00:00
51011	190164	Samsung Galaxy J5 (2016) LTE, zlatá	6	4990	CZK	22. 12. 2016 0:00:00
51011	191493	Mio Spirit 5400 - 4,3", CZ/SK, Lifetime	10	1799	CZK	22. 12. 2016 0:00:00
51011	193664	ASUS ZenWatch 2 Wren, stříbrná, pásek khaki	2	3990	CZK	22. 12. 2016 0:00:00
13211	190684	Lenovo B50-50, černá	5	8990	CZK	1. 12. 2016 0:00:00
13211	202138	Lenovo IdeaPad YOGA 510-14IKB, černá	7	12999	CZK	1. 12. 2016 0:00:00

Obrázek 4: Zpracování dat počáteční verze webového rozhraní

6 Časové náročnosti

Příprava na každé školení mi zabrala zhruba 4 hodiny. Tento čas byl věnován nastudování problematiky nebo zopakování a vytvoření vhodné prezentace. Doba samotné schůze s prezentací se pohybovaly ve vždy okolo dvou hodin.

Na projektu certifikátů jsem strávil 48 hodin, celkem dost času zabralo naučit se problematiku certifikátů a vymyslet strukturu aplikace. Zadání, které mi bylo dáno, je velice abstraktní a spousta věcí a požadavků v něm byla nejasná. V požadavcích byla i vygenerovaná dokumentace, najít vhodný způsob pro vytvoření dokumentace byl problém.

Projekt zpracování dat byl nejnáročnější, celková doba strávená nad projektem je asi měsíc. Projekt zabral hodně času z důvodu doučování se chybějících znalostí a častých změn požadavků nebo zadávání nových úkolů.

7 Využité a chybějící znalosti

7.1 Využité znalosti

Během školení na téma sql jsem využil znalosti získané v předmětech Úvod do databázových systémů a Databázové a informační systémy. Na školení se většinou probíraly věci, o kterých jsem již slyšel, setkal jsem se ovšem s pojmy příkaz merge a common table expression, které bylo nutné se doučit, a pro lepší pochopení jsem si vypracoval několik příkladů na tyto příkazy. Základní rozdíly mezi oracle a sql serverem jsem zjistil v předmětu Databázové a informační systémy II.

Školení z objektově orientovaného programování bylo víceméně opakování toho, s čím sem se již setkal při studiu. Základní postupy programování jsem se naučil v předmětu Algoritmy I a II, s principy fungování objektů a objektovým programováním jsem se setkal v předmětech Programování II, Programovací jazyky I a II.

Návrhové vzory byly probrány v předmětu Vývoj informačních systémů.

K užítku mi byly znalosti přístupu k databázi v kódu a návrhové vzory relačního mapování, které jsem získal v předmětu Databázové a informační systémy II při vytváření semestrálního projektu, které jsou dále rozvedeny a důkladněji probrány v předmětu Vývoj informačních systémů. Znalosti z tohoto předmětu jsem také uplatnil u projektu Zpracování dat, u tohoto projektu se mi také hodili poznatky z předmětu Úvod do softwarového inženýrství při analýze požadavků na projekt a postupu získání potřebných informací od zadavatele projektu a návrhu třídního diagramu projektu.

7.2 Chybějící znalosti

U projektu strom jsem využil znalostí získaných v předmětech Programovací jazyky. U projektu byl požadavek na vytvoření unit testů, což jsou funkce visual studia pro otestování funkcí. Unit testy spouští funkce a porovnávají, zda výsledek funkce je totožný s výsledkem, který specifikoval uživatel jako správný. S unit testy jsem se při studiu nesetkal a bylo nutné se je doučit.

Při vytváření projektu Služba Certifikátů jsem postrádal znalosti ohledně certifikátů, problematiku certifikátů jsem si musel sám dostudovat.

U projektu wizard mi scházeli znalosti, při úpravě webového rozhraní, ohledně technologie ASP.NET. S touto technologií jsem se setkal v předmětu Architektura technologie .NET, ale nebyla důkladně probrána.

Při projektu zpracování dat, mi scházeli znalosti ohledně tvorby webových stránek s použitím ASP.NET MVC. Tuto technologii jsem se musel doučit samostudiem. Také jsem neuměl pracovat s javascriptem, se kterým jsem nikdy nemusel pracovat, proto mi scházeli veškeré praktické i teoretické znalosti práce s ním.

8 Závěr

Na praxi jsem se naučil, co to znamená s někým spolupracovat na projektu a hlavně důklad na návrh struktury aplikace. Při implementaci byl kladen velký důraz na komentování kódu, u kterého nebylo na první pohled jasné, jakou vykonává činnost a také komentáře metod, které popisovali, k čemu metody slouží. Na závěr bych chtěl poděkovat kolegům, se kterými jsem měl možnost pracovat a našemu vedoucímu za to, že si dal tu práci, aby nás něco nového přiučil. Jsem taky rád, že na praxi bylo kritizováno a celkově hodnoceno styl jakým kód píšu, pomocí kritiky a správného navedení jsem se naučil základy, jak by měl správný kód vypadat a jak postupovat při psaní zdrojového kódu.